Introduction to Application Level Data Encryption

Max Averbukh

Northern Trust

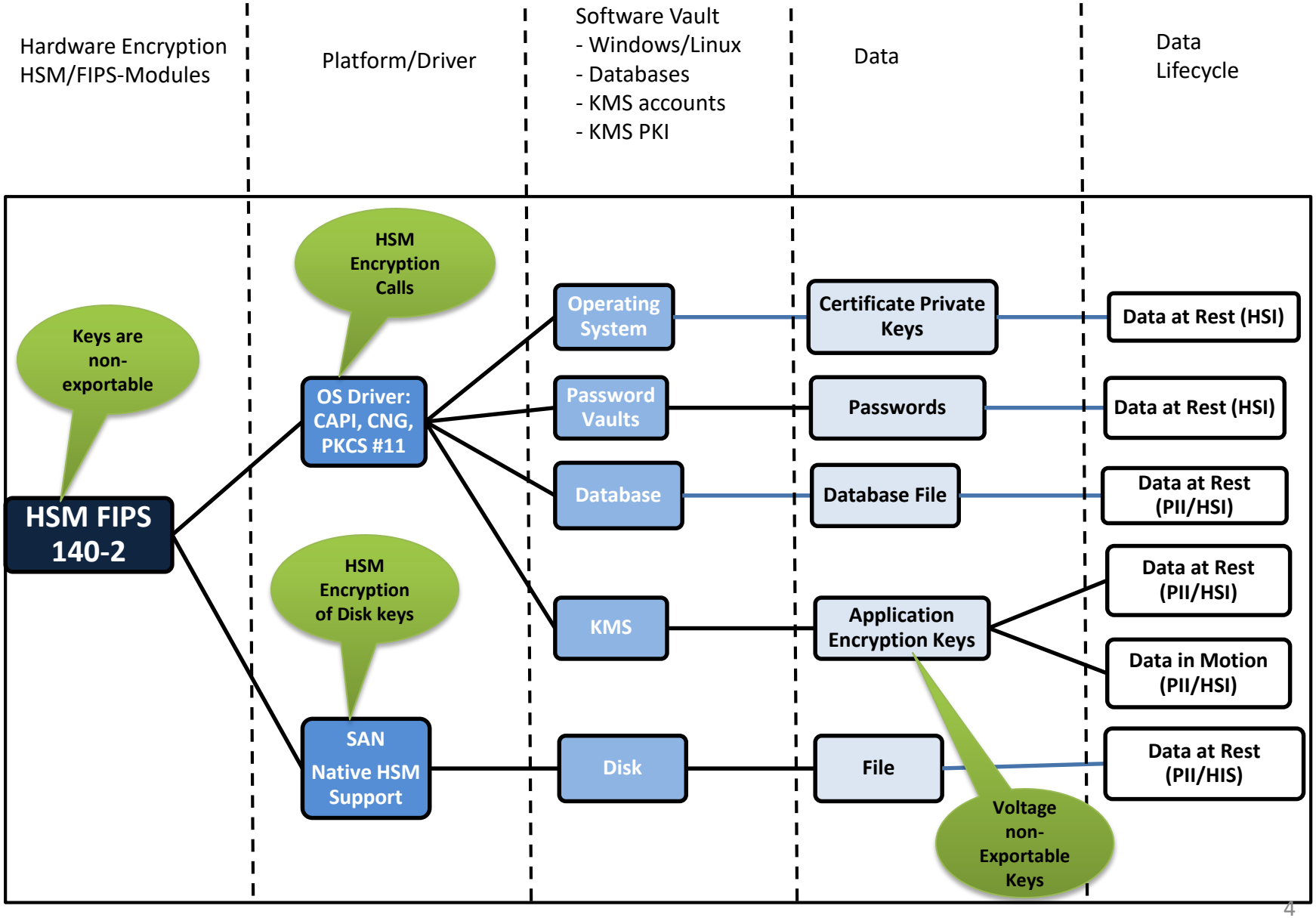
Objectives

- Explain available technologies within Application level data encryption.
- Understanding of the encryption layers-tree

Agenda

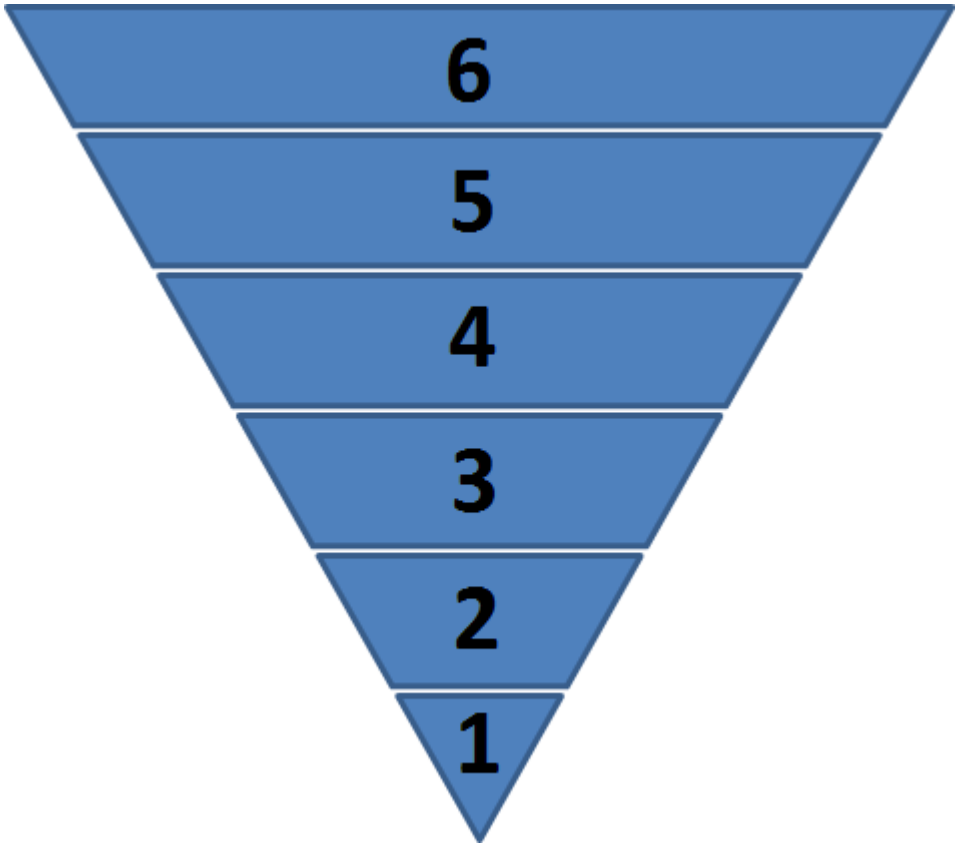
- Encryption Layers-Tree
- Hardware Level Protection
- Disk Level Protection
- Application and Database File Level Protection
- Application Field Level Protection
- Q&A

Encryption Layers-Tree



Simplified Layers View

6	Application Business Rules through Entitlements
5	Application Field level Encryption through Library, REST API or Kernel-driver
4	Application Field level Encryption through Network based Proxy
3	Whole File Encryption
2	Whole Disk and Drive Volume Encryption
1	Encryption Key Protection through HSM or KMS based on HSM

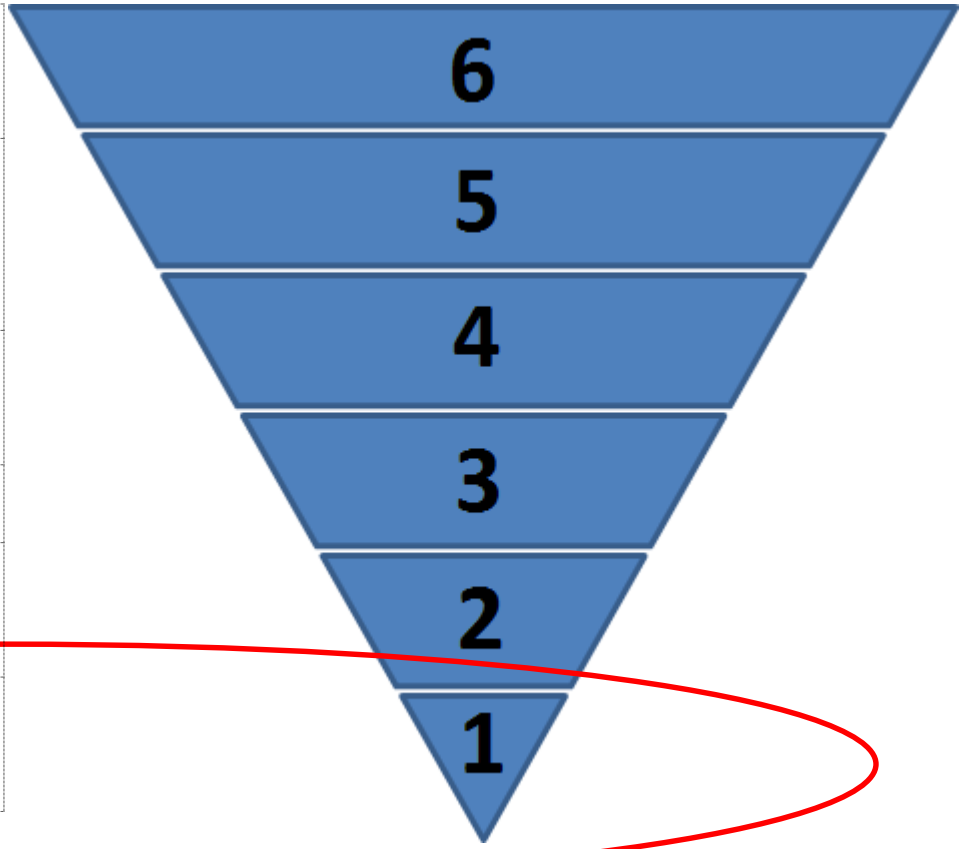


References

- Network path of data-centric layered approach link. This presentation and blog in the link were not based on each other and link is used as an example of what is out there: https://techbeacon.com/data-centric-security-changes-vulnerability-game
- Disk CWE: https://cwe.mitre.org/data/definitions/313.html

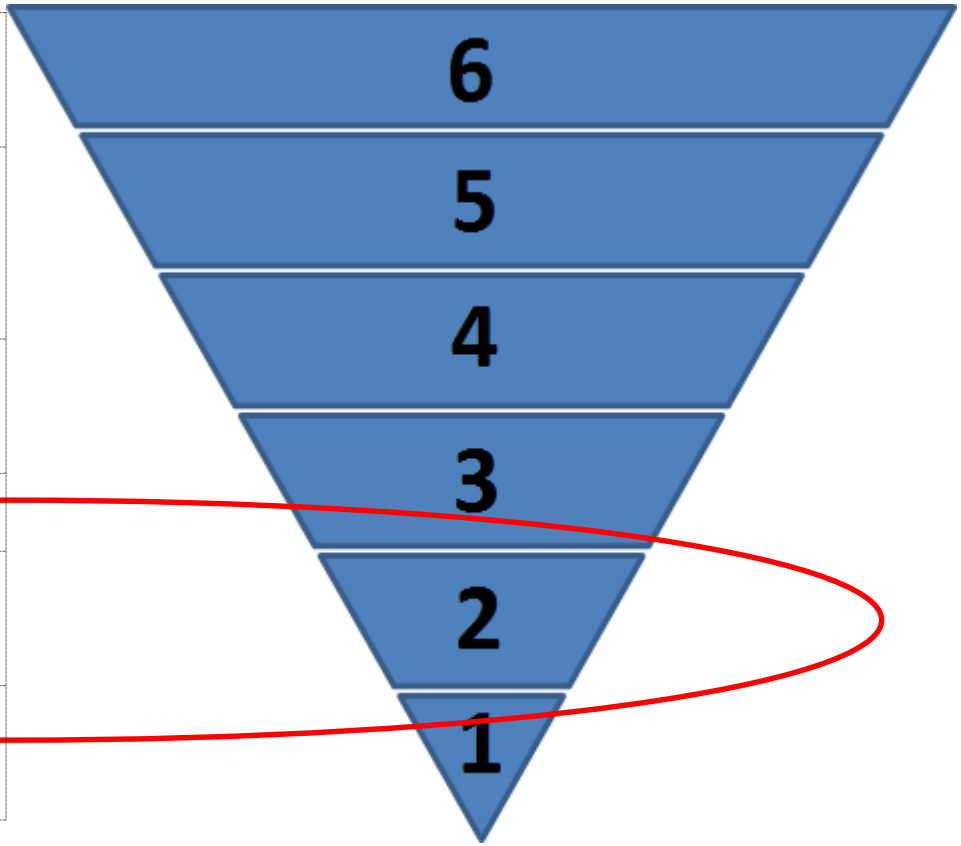
Hardware Level (1) Protection

6	Application Business Rules through Entitlements
5	Application Field level Encryption through Library, REST API or Kernel-driver
4	Application Field level Encryption through Network based Proxy
3	Whole File Encryption
2	Whole Disk and Drive Volume Encryption
1	Encryption Key Protection through HSM or KMS based on HSM



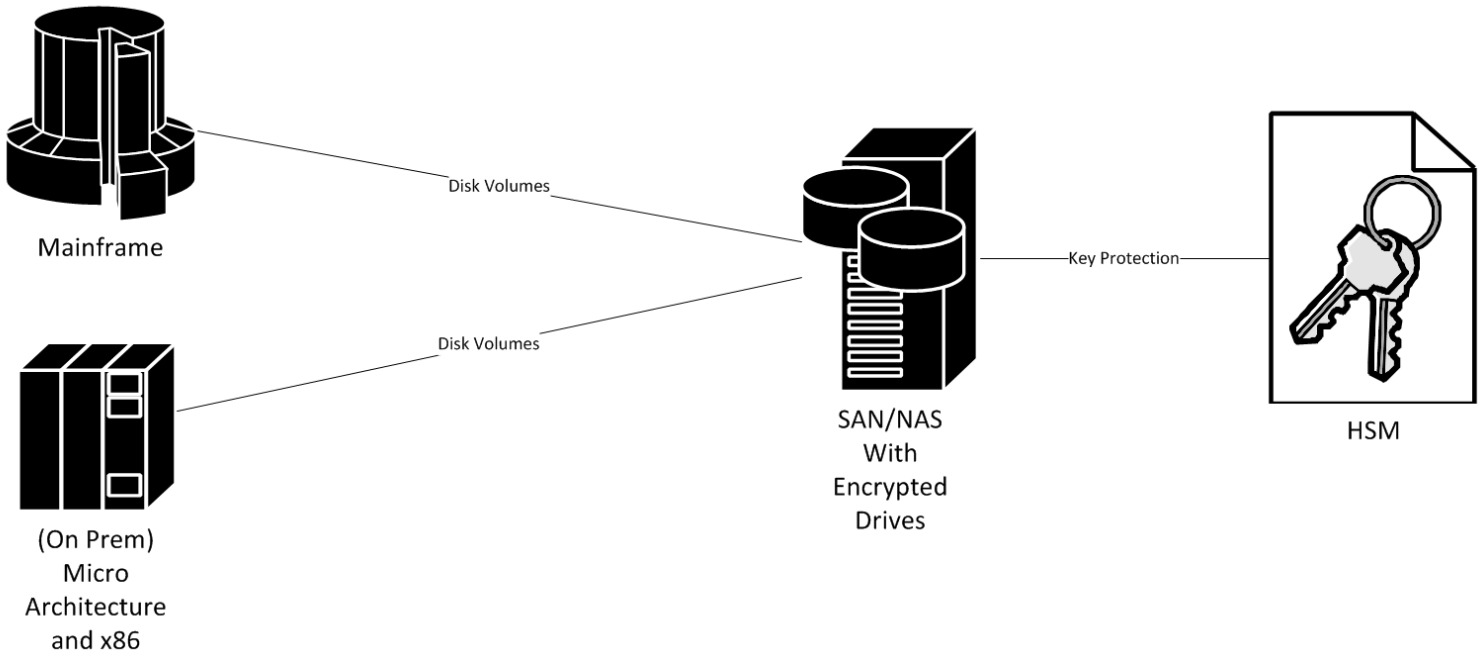
Disk Level (2) Protection

6	Application Business Rules through Entitlements
5	Application Field level Encryption through Library, REST API or Kernel-driver
4	Application Field level Encryption through Network based Proxy
3	Whole File Encryption
2	Whole Disk and Drive Volume Encryption
1	Encryption Key Protection through HSM or KMS based on HSM



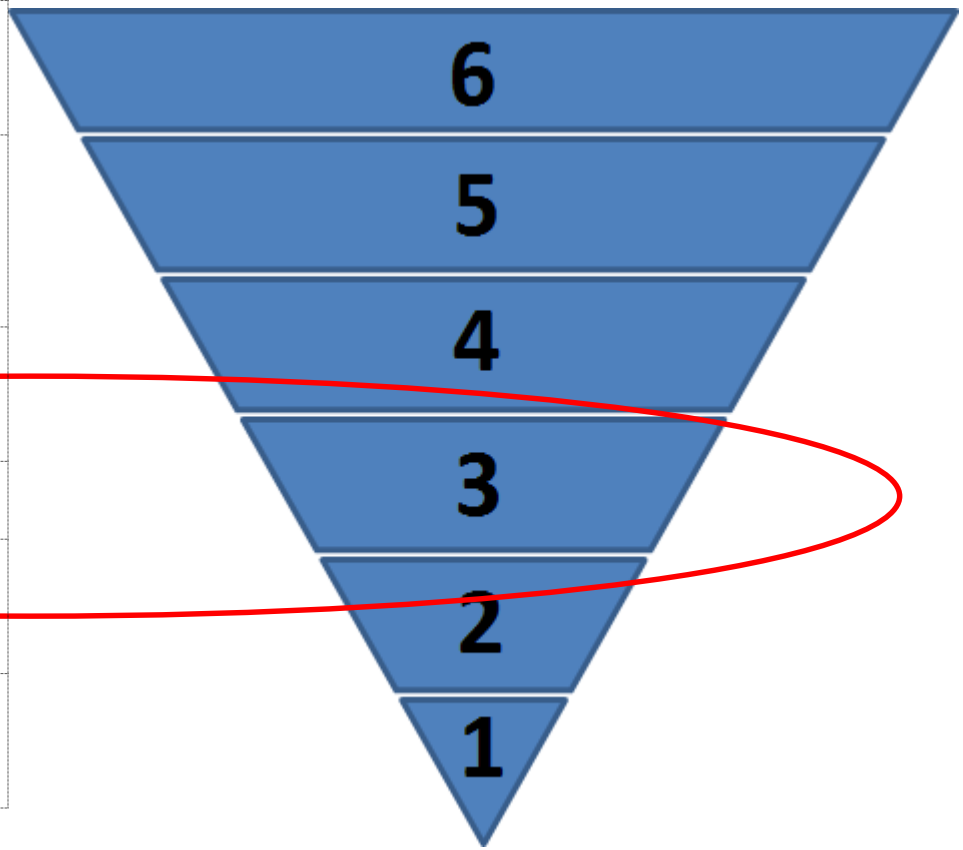
Disk Level (2) Protection

- Modern disks and controllers such as RAID, SAN and NAS can work in concert to enable modern encryption algorithms to protect all disk content.
- Here is an example of SAN storage volume encrypted with a key residing on HSM.
- Risk associated with partial (a disk loss) or full SAN decommissioning is removed because a SAN has to be on the same network as the key holding HSM for any of the drives to provide clear text file system.



Application and Database File Level (3) Protection

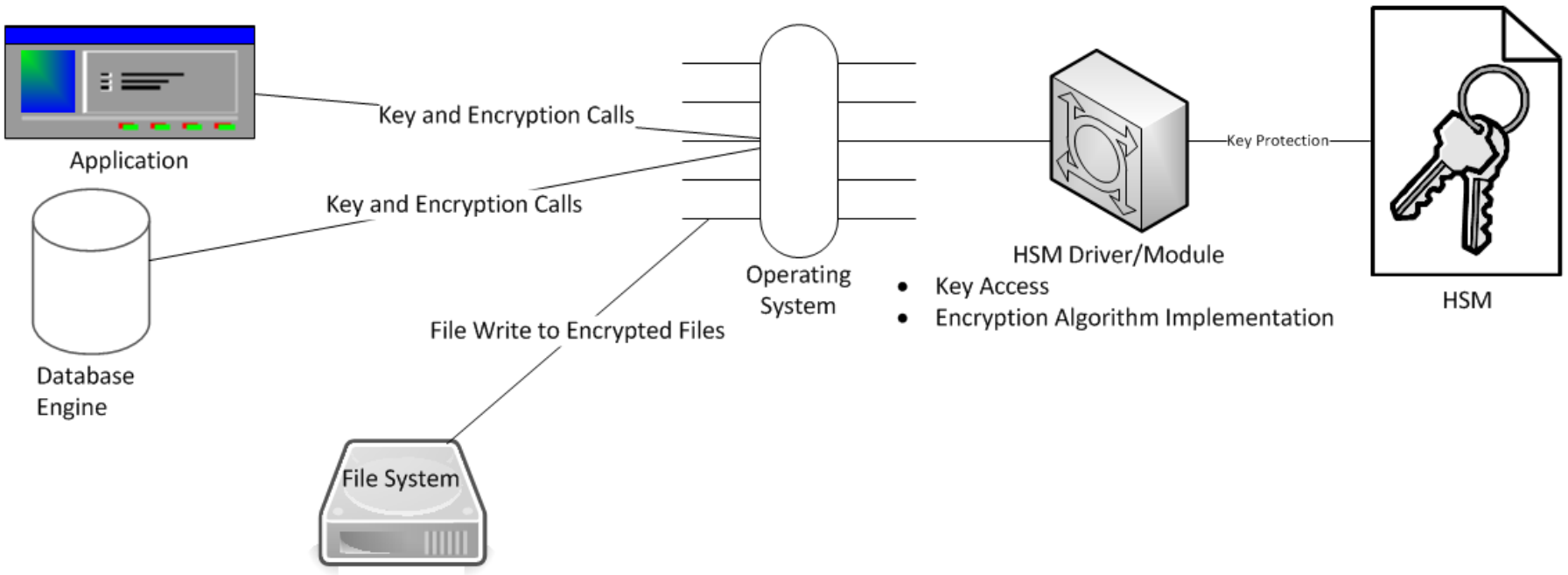
6	Application Business Rules through Entitlements
5	Application Field level Encryption through Library, REST API or Kernel-driver
4	Application Field level Encryption through Network based Proxy
3	Whole File Encryption
2	Whole Disk and Drive Volume Encryption
1	Encryption Key Protection through HSM or KMS based on HSM



Application and Database File Level (3) Protection

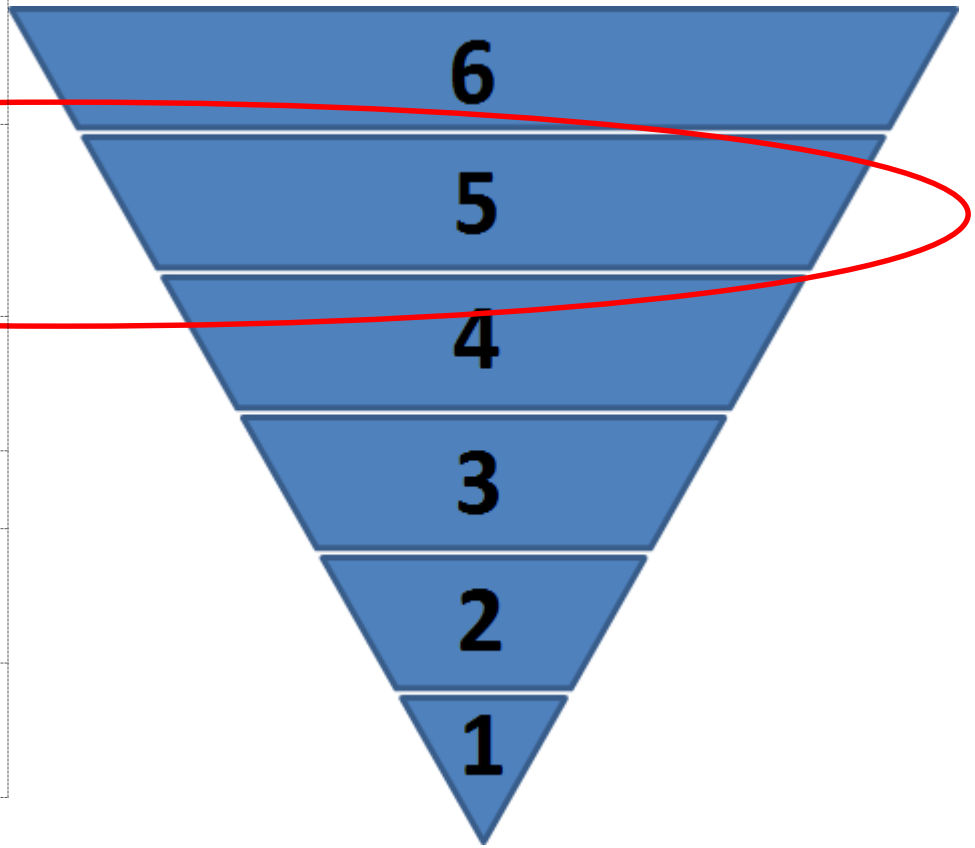
- Many of today's applications are capable of communicating to the operating system (OS/kernel) provided encryption drivers and modules
- HSM vendors allow for driver configuration to enable HSM key storage and various cryptographic services
- Here is an example of an application using a driver to encrypt its data storage files while writing to disk.
- It reduces risk associated with data confidentiality if these files are moved outside of this specific application services

File Level Protection, Application to HSM Encryption Architecture

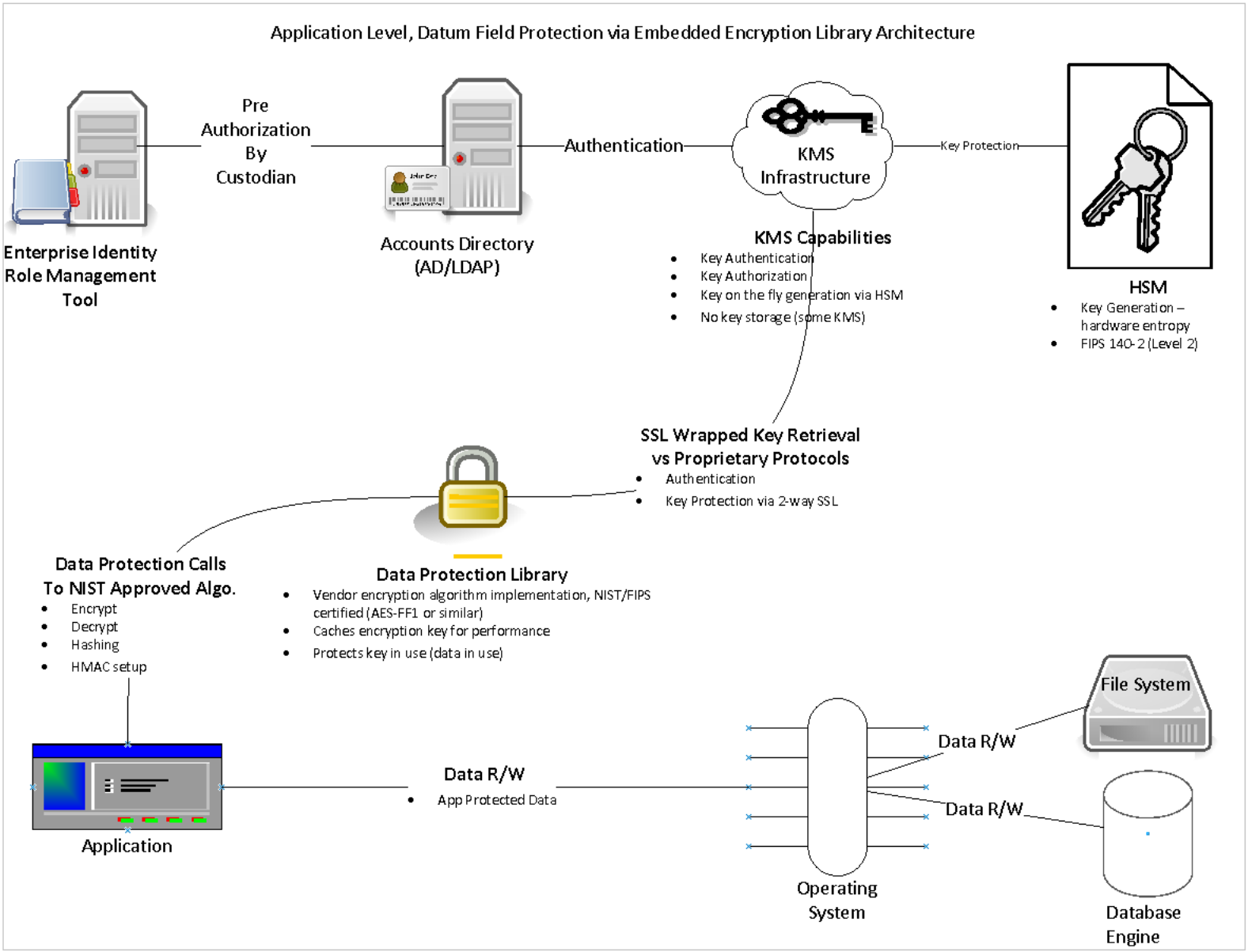


Application Field Level Protection Through an Integrated Library

6	Application Business Rules through Entitlements
5	Application Field level Encryption through Library, REST API or Kernel-driver
4	Application Field level Encryption through Network based Proxy
3	Whole File Encryption
2	Whole Disk and Drive Volume Encryption
1	Encryption Key Protection through HSM or KMS based on HSM



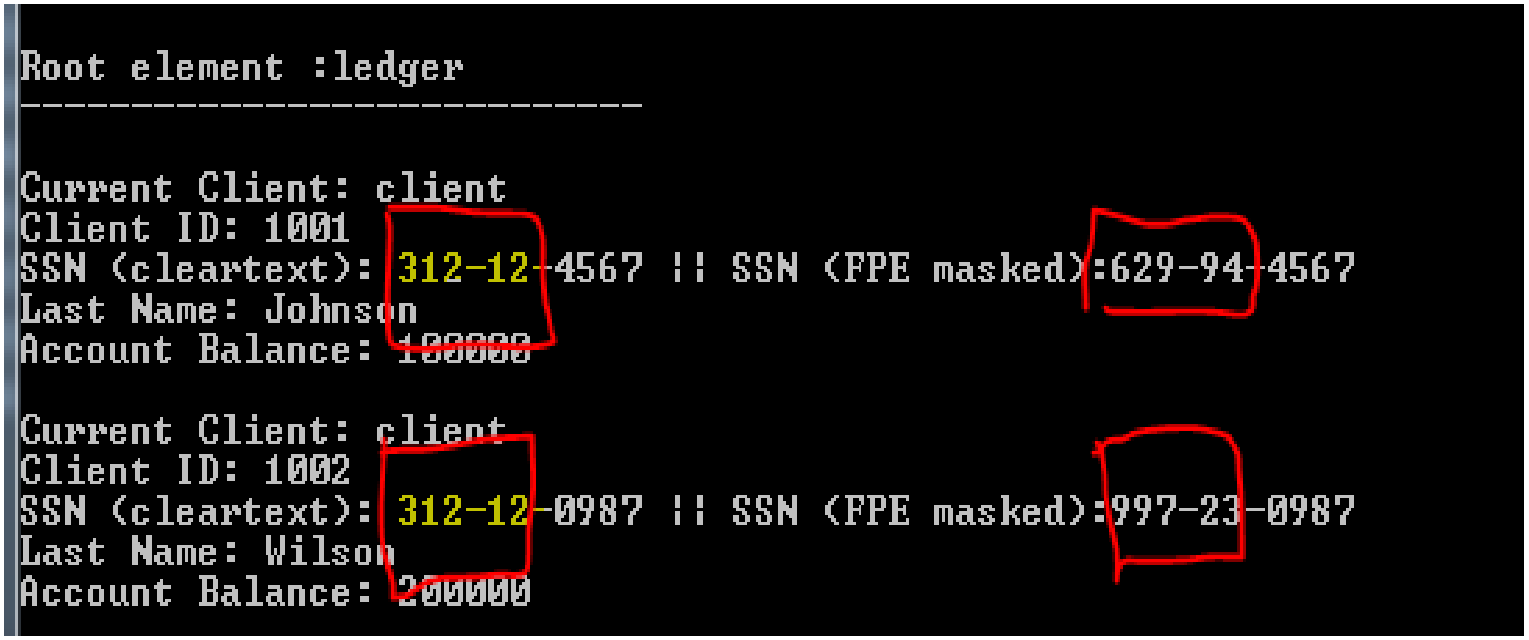
Application Field Level (5) Protection Through an Integrated Library



Code Example

```
//SSN_5P_4C
String  format = "SSN_5P_4C";
//Encryption Key Name
String  identity = "systest";

// Load the JNI library
System.loadLibrary("vibesimplejava");
LibraryContext library = null;
FPE            fpe     = null;

try {
    library = new LibraryContext.Builder().setPolicyURL(policyURL)
          .enableMemoryCache(true).setTrustStorePath(trustStorePath).setClientIdProduct(productName,
          productVersion).build();

    fpe = library.getFPEBuilder(format)
          .setSharedSecret(sharedSecret)
          .setIdentity(identity).build();

    File fXmlFile = new File("sample_data.xml");
    DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
    DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
    Document doc = dBuilder.parse(fXmlFile);
    doc.getDocumentElement().normalize();

    System.out.println("Root element :" + doc.getDocumentElement().getNodeName());
    NodeList nList = doc.getElementsByTagName("client");

    for (int temp = 0; temp < nList.getLength(); temp++) {
        Node nNode = nList.item(temp);
        System.out.println("\nCurrent Client: " + nNode.getNodeName());

        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element eElement = (Element) nNode;
            System.out.println("Client ID: " + eElement.getAttribute("id"));
            System.out.println("SSN (cleartext): "
                + eElement.getElementsByTagName("SSN").item(0).getTextContent()
                + " || SSN (FPE masked):"
                + fpe.protect(eElement.getElementsByTagName("SSN").item(0).getTextContent())
                );
```

Code Example Output



Application Field Level (5) Protection Through a REST API Web-Service

6 Application Business Rules through Entitlements

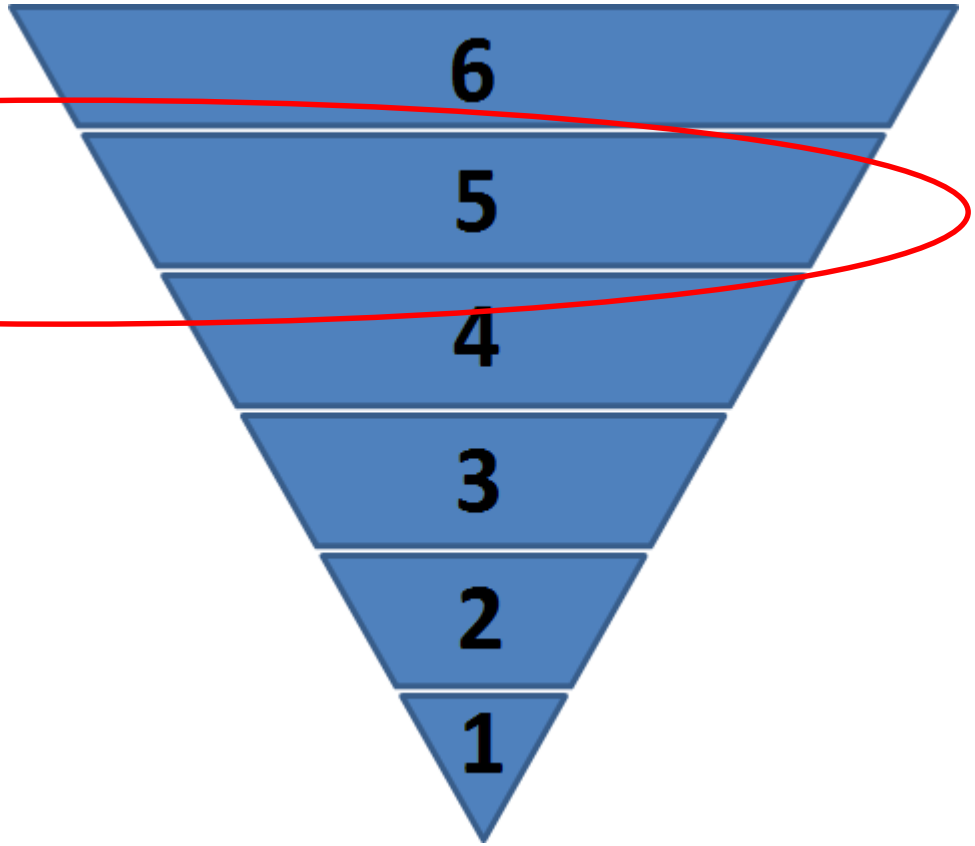
5 Application Field level Encryption through Library, REST API or host OS

4 Application Field level Encryption through Network based Proxy

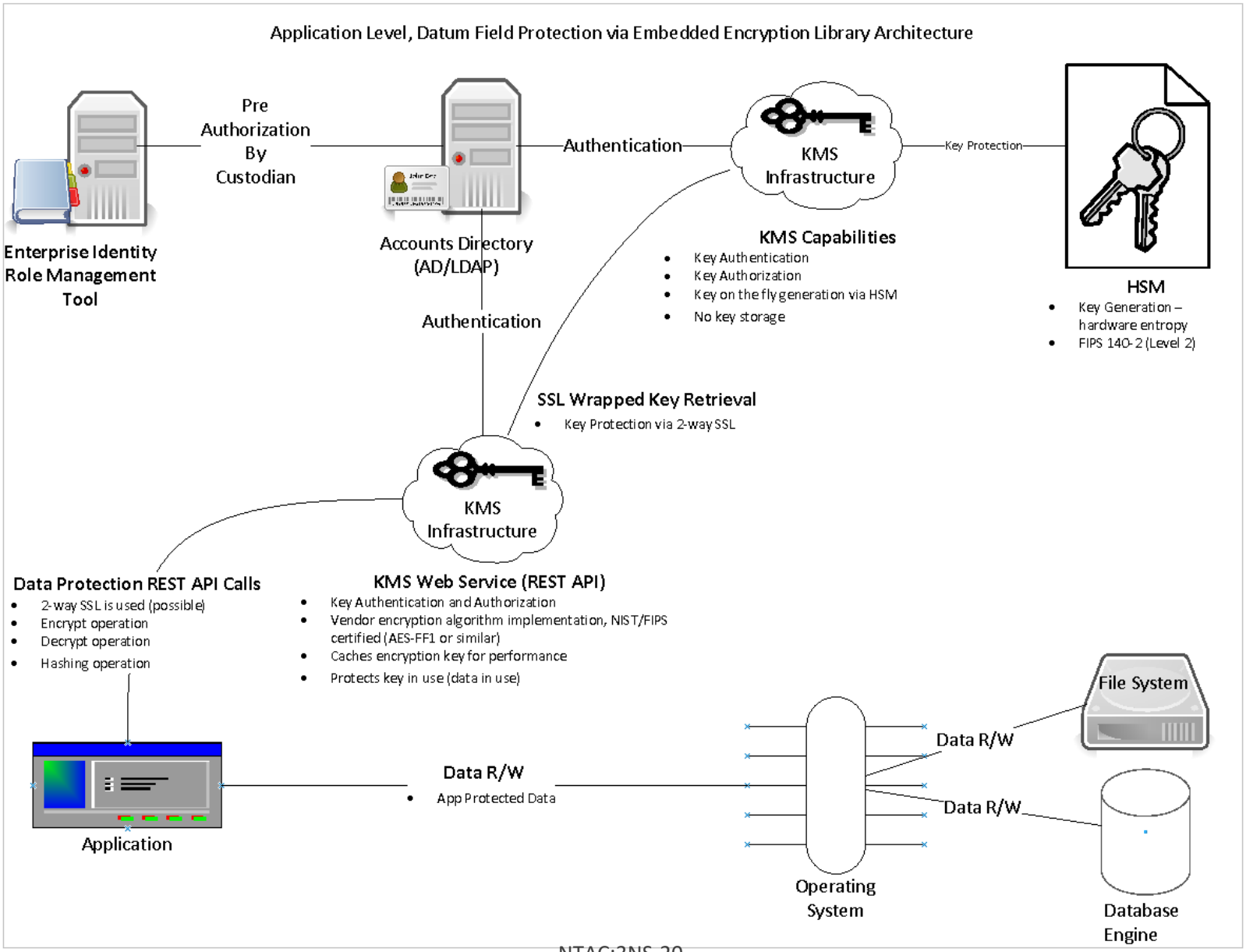
3 Whole File Encryption

2 Whole Disk and Drive Volume Encryption

1 Encryption Key Protection through HSM or KMS based on HSM

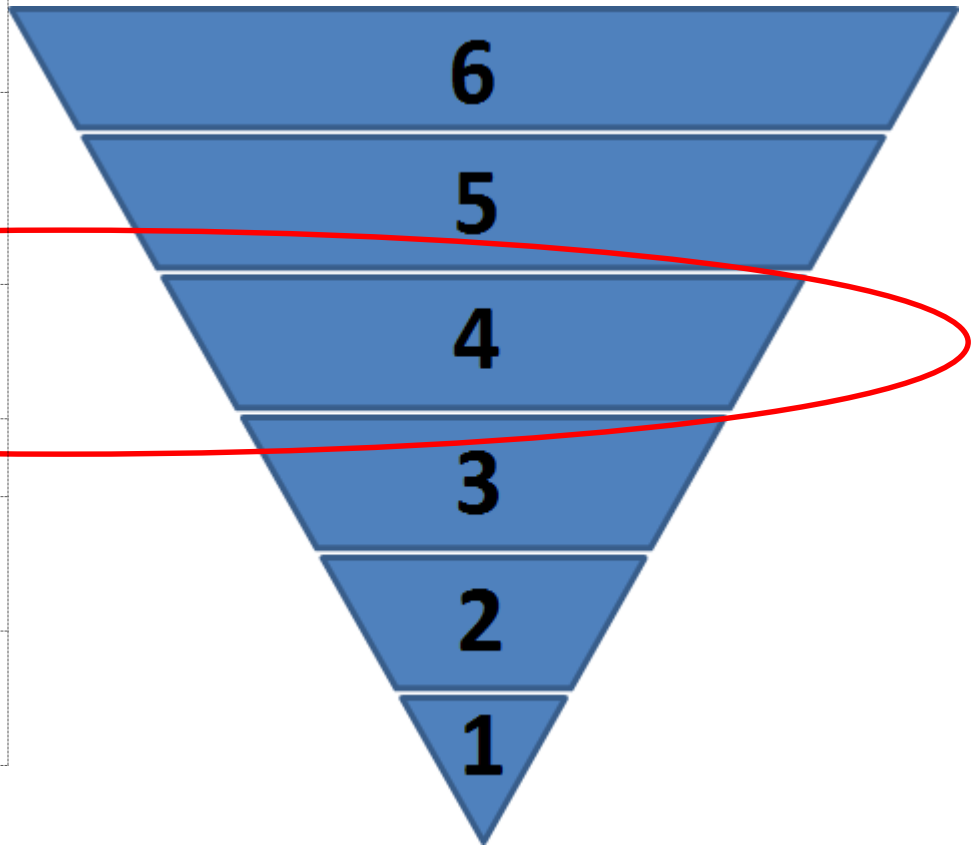


Application Field Level (5) Protection Through a REST API Web-Service



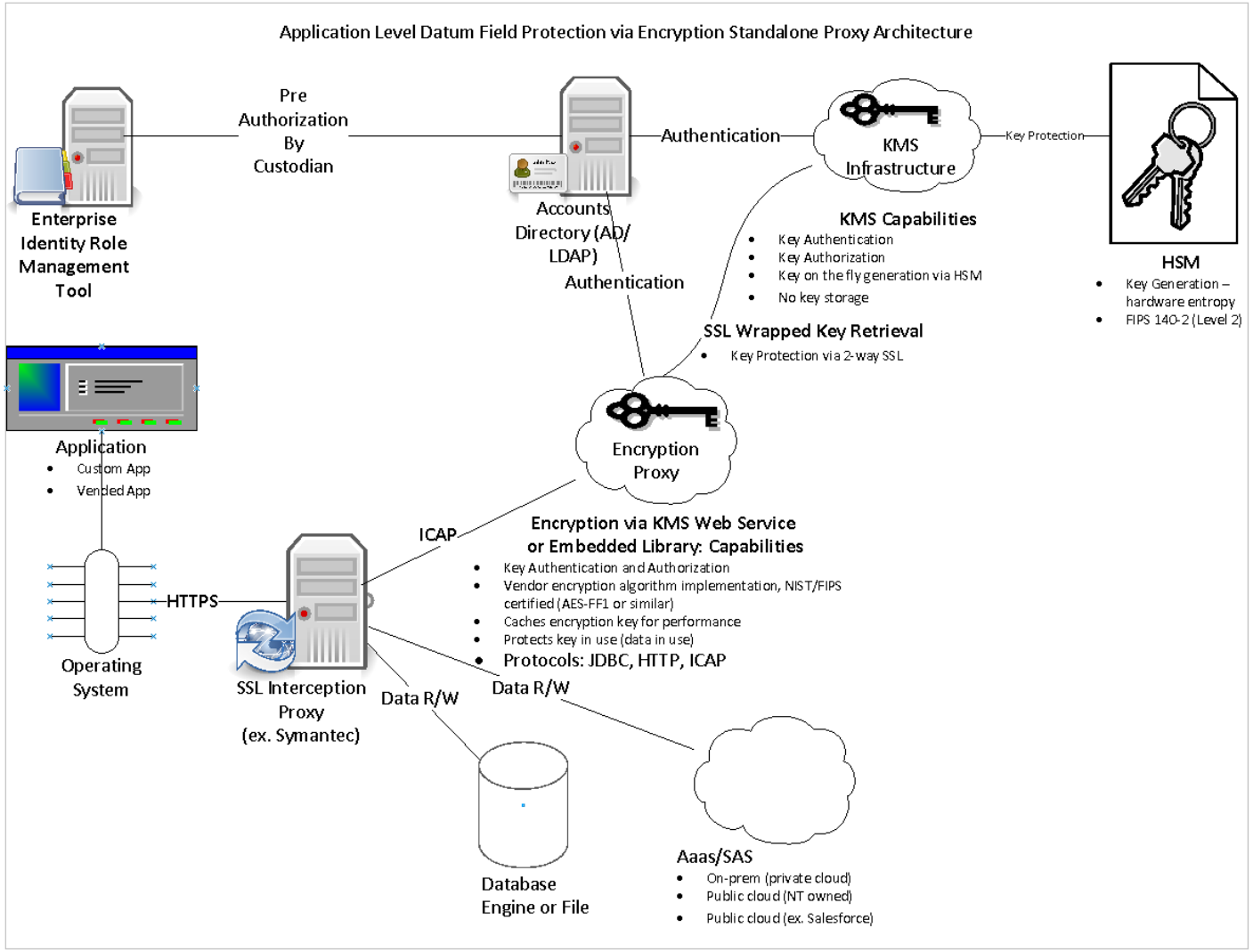
Application Field Level (4) Protection Through a Proxy-based Interception

6	Application Business Rules through Entitlements
5	Application Field level Encryption through Library, REST API or Kernel-driver
4	Application Field level Encryption through Network based Proxy
3	Whole File Encryption
2	Whole Disk and Drive Volume Encryption
1	Encryption Key Protection through HSM or KMS based on HSM



Application Field Level (4) Protection Through a Proxy-based Interception

- Data flows from an application and to the storage or cloud solution through a standard SSL terminating web proxy
- Network Proxy product operating in a DLP manner is configured to be inline with proxy rules to analyze traffic to identify fields for protection via encryption. It only talks to the web proxy and then the proxy forwards on data after it was protected through KMS.



Application Field Level Protection Other Topics

- Applications with built-in Field level protection for example: MongoDB, MarkLogic, etc.
- Vended applications requiring significant changes
- Public cloud environment integrations do include ability to apply the same layers, but have a number of other considerations
- Separation of Duty between keys ownership and application ownership
- Keys logical assignment across the enterprise for data sharing on common data-lakes

Q&A